Available online at www.sciencedirect.com
SCIENCE @ DIRECT®

An International Journal
computers & mathematics with applications

ELSEVIER Computers and Mathematics with Applications 50 (2005) 393–398
www.elsevier.com/locate/camwa

Queueing Network Models of Credit-Based Flow Control

M. EL-TAHA
Department of Mathematics and Statistics, University of Southern Maine
96 Falmouth Street, Portland, ME 04104-9300, U.S.A
eltaha@usm.maine edu

J. R. HEATH
Sun Microsystems, 1 Network Drive
Burlington, MA 01803, U S A.
John.Heath@sun.com

(Received December 2004, accepted April 2005)

Abstract—Credit-based flow control schemes are a commonly used means of preventing buffer overflow in high-speed networks spanning a local area Fibre Channel, a widely-used storage area networking technology, and InfiniBand, a recently developed system area network technology, are examples of two network technologies that employ credit-based flow control. With credit-based flow control, the receiver sends credits to the sender to indicate the availability of receive buffers, the sender waits for credits before transmitting messages to the receiver We present two models of credit-based flow control operation In particular, we consider a fork-join queueing system with two input queues, the message population feeds one queue and the credit population the other. We consider the case of bulk message arrivals and single arrivals drawn from a finite population Our analysis yields stationary probability distributions for message queue length and number of available credits We provide equations for mean message queue length, mean number of credits, throughput, and mean message waiting time © 2005 Elsevier Ltd. All rights reserved.

Keywords—Fork-join queues, Bulk-arrival process, Credit-based flow control, Performance evaluation, InfiniBand, Fibre Channel.

1. INTRODUCTION

Credit-based flow control management schemes are becoming increasingly popular in high-speed networks that span a local area. Fibre Channel [1], the technology used to realize storage area networks, employs credit-based methods for both end-to-end and link-level flow control. Credit-based methods are also used for link-level and end-to-end flow in InfiniBand [2], a new system area network technology for interconnecting processors, IO nodes, controllers, and adapters. Credit-based virtual circuit flow control has also been proposed as an efficient means of implementing flow-controlled ATM networks [3].

Credit-based flow control management operates as follows. Before the sender transmits packets, it receives credits from the receiving node. Credits sent by the receiving or destination node indicate the availability of buffer space at the receiver. Once the sender has received credits from

0898-1221/05/$ - see front matter Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX
doi:10 1016/j.camwa.2005.04 008

the receiver, it can transmit as many packets as it has credits. The sender maintains a variable, *credit-count*, to keep track of the number of credits it possesses. Following each transmission, the credit-count is decremented. The sender can continue transmitting packets to the receiver while credit-count is greater than zero, but must stop when credit-count becomes zero. The receiver sends a credit to the sender when a buffer becomes available. That is, for each packet it removes from a buffer. The receiver can respond with a single credit for each frame received and forwarded, as is done in the Fibre Channel link-level flow control protocol, or multiple credits can be accumulated and sent in a single response following the receipt of several packets, as is the case with the Fibre Channel and InfiniBand end-to-end flow control protocols. On receipt of a credit or credits from the receiver, the sender increments its credit-count by the number of credits received.

Sender-side credit flow control operates as follows. When a message is ready for transmission, it is placed in the message queue. If credit-count is greater than zero, the message is forwarded to the transmit queue for transmission, and credit-count is decremented. If, upon arriving at the message queue, credit-count is zero, the message remains in the message queue until a credit is received at which time it is forwarded to the transmit queue.

On the receiver-side, message arrivals are placed in a receive buffer and the host signaled. The host receives the signal, either via an interrupt or polling, and copies the message. After emptying the buffer, a credit is sent to the sender, indicating the receiver is ready for the next message.

The models developed in this paper describe the behavior of sender-side credit-based flow control. In particular, we model the interaction of the message queue and permission-to-send credits. We present two models of credit-based flow control operation. We consider a fork-join queueing system with two input queues; the message population feeds one queue and the credit population the other. We consider the case of bulk message arrivals, and single arrivals drawn from a finite population. Fork-join queueing models [4] have been used for the analysis of multiprocessor systems by dividing a job into multiple tasks that are executed concurrently on the processing units.

The remainder of this paper is organized as follows. The next section describes and analyzes a fork-join queueing system with bulk message arrivals. We derive the stationary probability distribution message queue length and credit-count, and provide solutions for mean number of waiting messages, mean credit-count, throughput, and mean message waiting time. In Section 3, we consider a finite-source queueing system with both messages and credits drawn from finite populations. The system is analyzed to yield stationary probabilities and performance metrics. In the final section, we present an example to demonstrate the application of the paper's results.

2. BULK PACKET ARRIVAL MODEL

Consider a fork-join queueing model consisting of two input queues: the message queue, B_1 and, the credit-count, B_2. The queues are fed by arrivals from two populations: the sending host's drivers and the receiver's buffer manager. The size of the message population is infinite; the size of the receiver buffer population is K. The first population feeds the infinite buffer message queue, B_1, and the second population feeds the credit-count, B_2. As soon there is an object in each input queue, a message immediately departs the message queue, B_1, and the credit-count, B_2, is decremented by one. Therefore, at least one queue is always empty, and any objects in the other queue wait for an arrival to the other.

We solve this model to determine mean number of waiting messages, mean message queue waiting time, message throughput, mean number of credits. We also characterize the message queue departure process, and investigate the effect of the number of message buffers available at the receiver.

Messages arrive at B_1 according to a Poisson batch arrival process. Let the time until each batch in population one, a transmit message, requests service by joining the message queue, B_1, be exponential with parameter λ, and let the batch size, X, be a discrete probability distribution with pdf $C(x)$, $x \geq 1$, $\sum_{x=1}^{\infty} C(x) = 1$. Credits are sent one at a time. Let the time until each member in population two, a credit, requests service by joining the credit-count queue, B_2, be exponential with parameter μ. Then, the times between sending credits are exponential with parameters $n\mu$, where n is the number of elements remaining in the receiver's credit population. That is, if L is the number of credits in B_2, $n = K - L$

2.1. Model Analysis

Let $X_i(t)$ be the number of objects in queue B_i at time t, $i = 1, 2$. Then, $X(t) = X_1(t) - X_2(t) + K$ is a birth-death process with state space $S = \{0, \dots, \infty\}$. We assume that the system is stable. That is, $\lambda E[X]/(K\mu) < 1$. Next, we solve the balance equations to obtain the stationary distribution, $p(n) = \lim_{t\to\infty} P(X(t) = n)$, $0 \leq n \leq \infty$.

The flow balance equations are

$$\lambda p(0) = \mu p(1),$$

$$(\lambda + n\mu) p(n) = (n+1) \mu P(n+1) + \lambda \sum_{i=1}^{n} C(i) p(n-i), \qquad 1 \leq n < K,$$

$$(\lambda + K\mu) p(n) = K\mu p(n+1) + \lambda \sum_{i=1}^{n} C(i) p(n-i), \qquad n \geq K.$$

Solving the flow balance equations leads to the following recursive equations for evaluating the stationary distribution, see [5]

THEOREM 2.1. *The stationary probabilities are given by*

$$p(n) = G(n) p(0), \qquad n = 0, 1, 2, \dots,$$

where

$$A_j = \sum_{m=j}^{\infty} C(m),$$

$$a(n) = \lambda / \min(n, K)\mu,$$

$$G(0) = 1,$$

$$G(n) = a(n) \sum_{i=0}^{n-1} A_{n-i} G(i), \qquad n = 1, 2, \dots,$$

$$p(0) = \left[\sum_{n=0}^{\infty} G(n)\right]^{-1}.$$

2.2. Performance Metrics

Let L_1 be the mean number of objects in buffer B_1 and L_2 be the mean number of objects in buffer B_2. Moreover, let λ_e be the combined arrival rate to both queues which equals twice system throughput. Then, by definition,

$$L_1 = \sum_{j=1}^{\infty} j p(K+j),$$

$$L_2 = \sum_{j=0}^{K} (K-j) p(j),$$

$$\lambda_e = \lambda E[X] + \sum_{j=0}^{K} \min(j, K) \mu p(j).$$

Analytical formulas for L_2 and λ_e can be obtained using simple manipulations. Noting system stability and that packets and credits leave in pairs, we conclude that the effective arrival rates to both buffers are equal, thus,

$$\lambda_e = 2\lambda E[X] \qquad \text{and} \qquad \lambda E[X] = \mu(K - L_2).$$

Therefore,

$$L_2 = K - \lambda E[X]/\mu.$$

It also follows that the system message throughput is $X = \lambda_e/2 = \lambda E[X]$, and the mean time a message waits for a credit is

$$W_1 = 2L_1/\lambda_e.$$

3. FINITE PACKET-POPULATION MODEL

Consider a fork-join queueing model consisting of two finite-input queues, the message queue, B_1, and the credit-count, B_2. The queues are fed by arrivals from two finite populations,

(1) the message population and
(2) the receiver's buffer pool.

The size of the message population is K_1, and the size of the receiver buffer population is K_2. The first population feeds the message queue, B_1, and the second population feeds the credit-count, B_2. As noted above, as soon there is an object in each input queue, a message immediately departs the message queue, B_1, and the credit-count, B_2, is decremented by one. Therefore, at least one queue is always empty, and any objects in the other queue wait for an arrival to the other.

Let the time until each member in population one, a transmit message, requests service by joining the message queue, B_1, be exponential with parameter λ_1. Let the time until each member in population two, a credit, requests service by joining the credit-count queue, B_2, be exponential with parameter λ_2. Then, the times between requests are exponential with parameters $n_i\lambda_i$, where n_i is the number of elements in population i not queued, i.e, if L_i is the number of elements queued in B_i,

$$n_i = K_i - L_i, \qquad i = 1, 2.$$

3.1. Model Analysis

Let $X_i(t)$ be the number of objects in queue B_i at time t, $i = 1, 2$. Then, $X(t) = X_2(t) - X_1(t)$ is a birth-death process with state space $S = \{-K_1, \dots, 0, \dots, K_2\}$. The transition rates are given by

$$q(i, i+1) = \begin{cases} K_2\lambda_2, & -K_1 < i \le 0, \\ (K_2 - i)\lambda_2, & 0 \le i \le K_2 - 1, \end{cases}$$
$$q(i, i-1) = \begin{cases} (K_1 - i)\lambda_1, & -K_1 + 1 \le i \le 0, \\ K_1\lambda_1, & 0 \le i \le K_2. \end{cases}$$

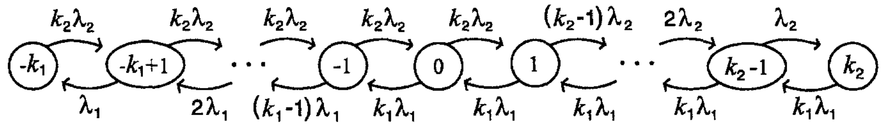

Figure 1 Flow balance diagram

The state transition diagram is shown in Figure 1

The stochastic process $\{X(t), t \geq 0\}$ is a stable, finite-state birth-death process with $\lambda(i) = q(i, i+1)$ and $\mu(i) = q(i, i-1)$. The birth-death balance equations are solved to obtain the stationary distribution $p(n) = \lim_{t\to\infty} p(X(t) = n)$, $-K_1 \leq n \leq K_2$,

$$p(n) = \begin{cases} \dfrac{K_2^{n+K_1}}{(n+K_1)!} \left(\dfrac{\lambda_2}{\lambda_1}\right)^{n+K_1} p(-K_1), & -K_1 \leq n \leq 0, \\ \dfrac{K_2! K_2^{K_1}}{(K_2-n)! K_1! K_1^n} \left(\dfrac{\lambda_2}{\lambda_1}\right)^{n+K_1} p(-K_1), & 1 \leq n \leq K_2, \end{cases} \tag{1}$$

$$p(-K_1) = \left[1 + K_2^{K_1} (\lambda_2/\lambda_1)^{K_1} \sum_{n=-K_1+1}^{0} \frac{K_2^n}{(n+K_1!)} \left(\frac{\lambda_2}{\lambda_1}\right)^n \right.$$
$$\left. + \frac{K_2! K_2^{K_1}}{K_1!} (\lambda_2/\lambda_1)^{K_1} \sum_{n=1}^{K_2} \frac{1}{(K_2-n)! K_1^n} \left(\frac{\lambda_2}{\lambda_1}\right)^n \right]^{-1},$$

where $p(-K_1)$ is the normalizing constant.

3.1. Performance Metrics

Let L_1 and L_2 be the mean number of objects in queues B_1 and B_2, respectively. Moreover, let λ_e be the combined arrival rate to both queues which equals twice system throughput. Then,

$$\begin{aligned} L_1 &= \sum_{n=-K_1}^{-1} -np_n = \sum_{n=1}^{K_1} np(-n), \\ L_2 &= \sum_{n=1}^{K_2} np_n, \\ \lambda_e &= \sum_{n=-K_1}^{-1} q(n, n+1) p_n + \sum_{n=1}^{K_2} q(n, n-1) p_n, \\ &= K_2\lambda_2 \sum_{n=-K_1}^{-1} p_n + K_1\lambda_1 \sum_{n=1}^{K_2} p_n. \end{aligned}$$

Simple analysis will allow us to compute some performance measure in terms of others. More specifically, let λ_{ei}, $i = 1, 2$ be the effective arrival rate to buffer B_i. Now, we can easily show that

$$\begin{aligned} \lambda_{ei} &= (K_i - L_i)\lambda_i, \qquad i = 1, 2, \\ \lambda_e &= 2(K_1 - L_1)\lambda_1 = 2(K_2 - L_2)\lambda_2, \\ L_2 &= K_2 - \lambda_1 (K_1 - L_1)/\lambda_2. \end{aligned}$$

Let X^{-1} is the mean time between departures and λ_e^{-1} is the mean time between arrivals to buffers. Note that customers arrive one at a time, but depart two at a time. System stability implies $\lambda_e = 2X$. Moreover, the mean time spent in buffer B_i, W_i, per object is computed using $W_i = L_i/(K_i - L_i)\lambda_i$, $i = 1, 2$.

4. EXAMPLE

We apply the results of Section 2, bulk packet arrival model, to a specific system. To demonstrate the effect of number of receive buffers on performance, we compare waiting times and queue lengths as the number of receive buffers ranges from two to 16. Specifically, for a fixed

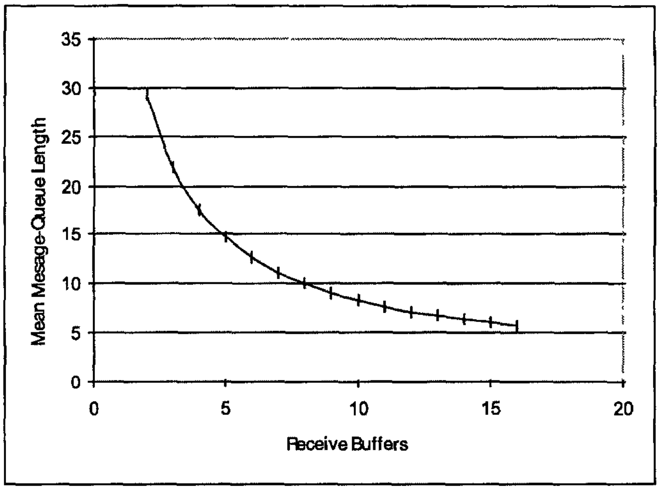

Figure 2. Mean message queue length, in packets, as the number of receive buffers ranges from two to 16

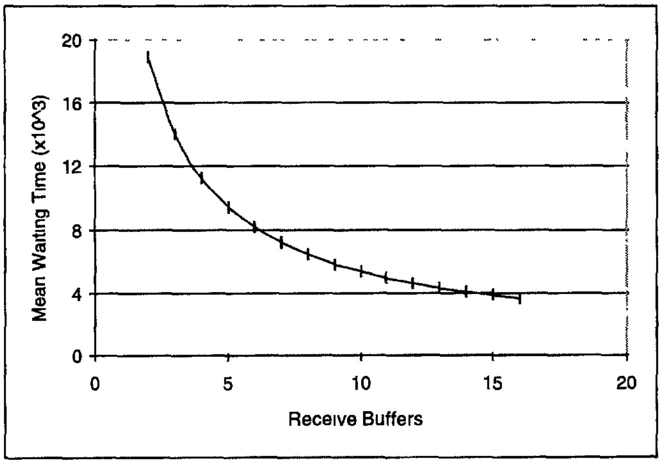

Figure 3 Mean packet waiting time, multiplied by 10^3, as the number of receive buffers ranges from two to 16

packet arrival rate, we derive the effect of changing receive buffer count on the mean number of packets in the message queue, Figure 2, and on the mean packet waiting time in the message queue, Figure 3. We assume the bulk packet arrival model with $\lambda = 100$ and $\mu = 6000$. Also, $C(x)$ is discrete uniform over $\{1 \ldots 32\}$ with $C(x) = 1/32$ over its domain.

REFERENCES

1. *Fibre Channel Physical and Signalling Interface (FC-PH), X3 230*, (1994).
2. *Infiniband*$^{\text{TM}}$ *Architecture Release 1.1 Volume 1—General Specifications*, (2002).
3. T Blackwell *et al*, Credit-based flow control for ATM networks, *IEEE Network*, 40–48, (March/April, 1995)
4. E. Varki and L.W. Dowdy, Analysis of balanced fork-join queueing networks, Proceedings of *The 1996 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, 232–241, (1996)
5. M L. Chaudhry and J G C Templeton, *A First Course in Bulk Queues*, John Wiley, New York, (1983).